Assigning values to objects

```
> x = 11
```

print(x) or just type x

```
> y <- 7
```

用箭頭或是等號都可以用來assign
也可以在寫其他行來取代
像是 > y <- 9，y 就會是代表9

若是想要刪除儲存的值 可以使用 -> rm(x)
就會查不到x對應的值

Basic arithmetic functions

😇Xy 可以拿來做基礎運算

```
> x+y
>x-y
> x*y
> x/y
> x^2
> y^2
> x^2 + y^2
> sqrt(y)
> y^(1/2)
> (log)y
```

😇注意：
當輸入的要求不完整時，輸出的值會顯示： +

```
> sqrt(y     <<少加了一個括號
+            <<
```

上述都是儲存numeric的型態

😊若是想要儲存成**character**的型態，則可以用**” “**引號來包住想要輸入的值

```
> xx <- “marin”
>  xx
[1] “marin”
```

Creating vectors, matrices

😇用c(.)來儲存vector

```
>  x <- 11
>  x
[1] 11

> x1 <- c(1,3,5,7,9)
> x1
[1] 1,3,5,7,9
```

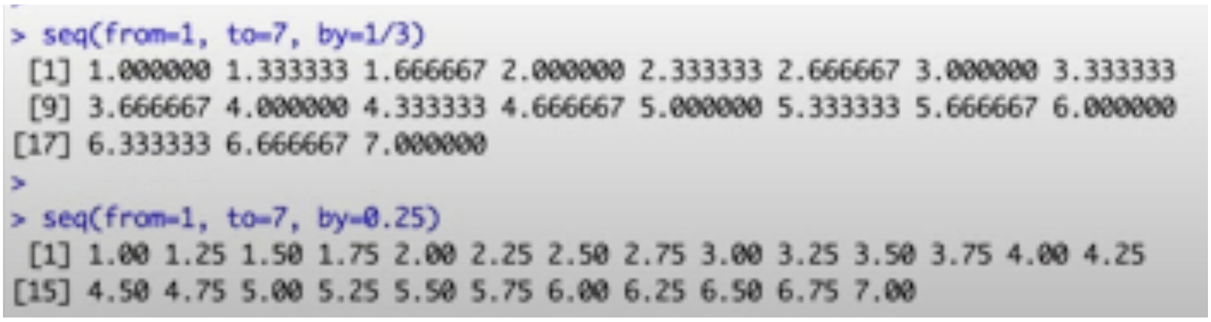
```
> gender <- c(“male”, “female”)
> gender
[1] “male”  “female”
```

😇用「:」來生成序列

```
>  2:7
[1] 2 3 4 5 6 7
```

也有另一種表示方法

```
> seq(from=1, to=7, by= 1)
[1] 2 3 4 5 6 7
```



```
> seq(from=1, to=7, by=1/3)
 [1] 1.000000 1.333333 1.666667 2.000000 2.333333 2.666667 3.000000 3.333333
 [9] 3.666667 4.000000 4.333333 4.666667 5.000000 5.333333 5.666667 6.000000
[17] 6.333333 6.666667 7.000000
>
> seq(from=1, to=7, by=0.25)
 [1] 1.00 1.25 1.50 1.75 2.00 2.25 2.50 2.75 3.00 3.25 3.50 3.75 4.00 4.25
[15] 4.50 4.75 5.00 5.25 5.50 5.75 6.00 6.25 6.50 6.75 7.00
```

😇當要產生重複出現的數值或字串時，可以使用rep (x, times=…)

```
> rep(1, times= 10)
[1] 1 1 1 1 1 1 1 1 1 1
```

```
> rep(“marin” ,times= 5)
[1] marin marin marin marin marin
```

```
>rep( 1:3, times=3)
[1] 1 2 3 1 2 3 1 2 3 1 2 3 1 2 3
```

向量的運算

```
> x = 1:5
> x
[1] 1 2 3 4 5
```

```
> y = c(1,3,5,7,9)
> y
[1] 1 3 5 7 9
```

可以對整行向量進行運算

```
> x +10
[1] 11 12 13 14 15
```

```
> x*10
```

```
[1] 10 20 30 40 50
```

如果兩個向量有同樣的length, 就可以進行加減乘除

```
> x+y
[1] 2 5 8 11 14
```

```
> x*y
> x/y 都可以進行運算
```

😇取出特定值 使用[ ]

```
> y[3] 取出y vector裡的第三個數值時
[1] 5
```

```
> y[-3]  加上負號之後, 就會取出除了第三個以外其他的值
[1] 1 3  7 9
```

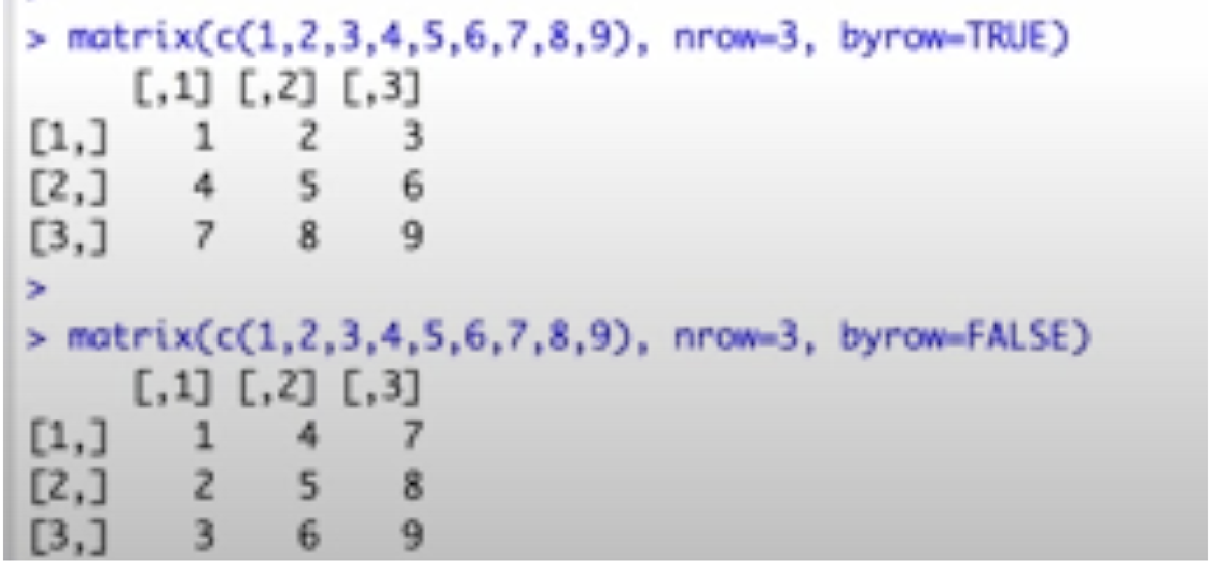
```
> y[1:3] 會取出前三個值
> 1 3 5
```

```
> y[c(1,5)] 搭配 vector就可以取出指定位置
[1] 1 9
```

```
> y[-c(1,5)] 加上負號, 就可以取出除了第一個跟第五個位置的向量剩餘數值
[1]   3 5 7
```

```
> y[ y<6] 也可以加上比較運算元進行比較後再輸出
[1] 1 3 5
```

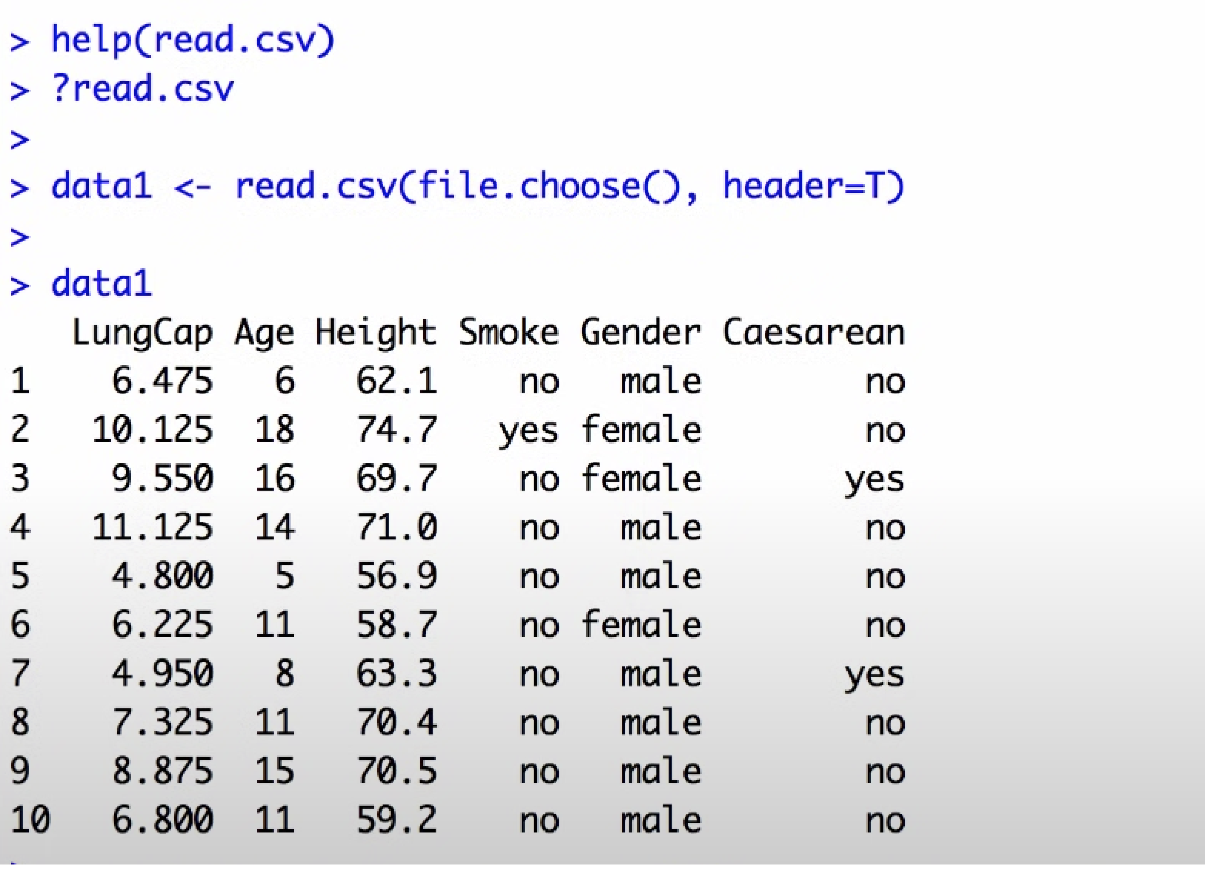
```
> matrix(c(1,2,3,4,5,6,7,8,9), nrow=3, byrow= TRUE)
```



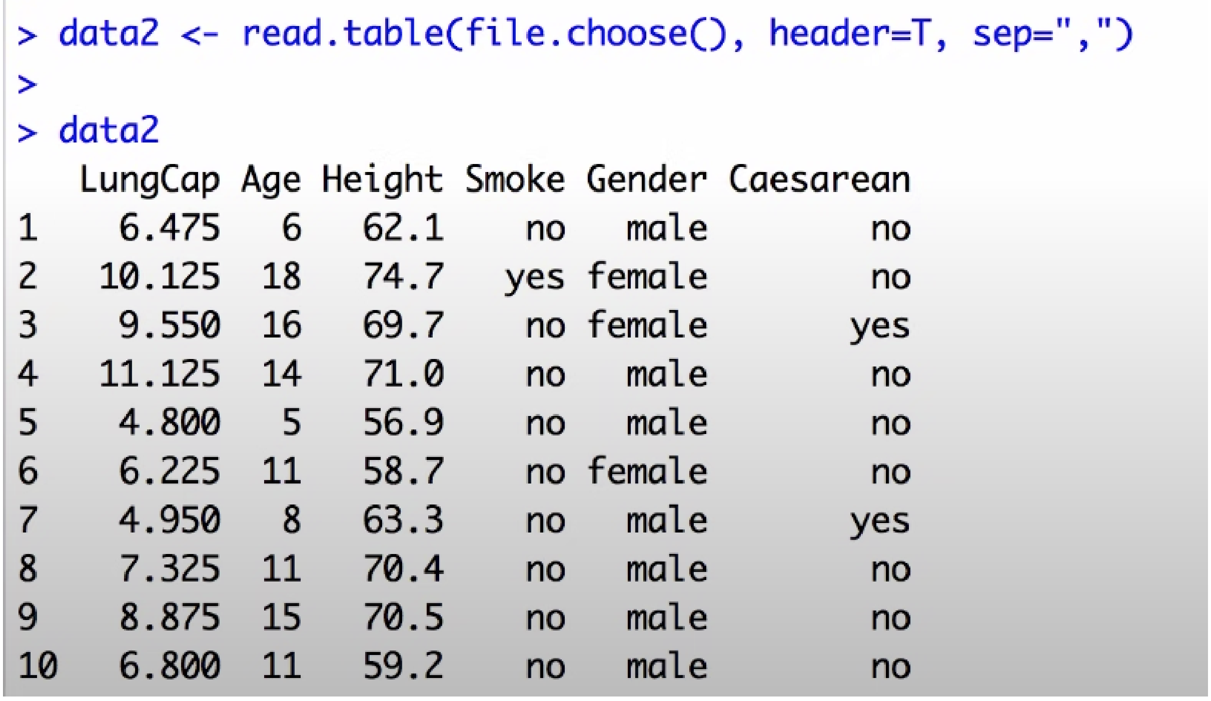
```
> matrix(c(1,2,3,4,5,6,7,8,9), nrow=3, byrow=TRUE)
     [,1] [,2] [,3]
[1,]    1    2    3
[2,]    4    5    6
[3,]    7    8    9
>
> matrix(c(1,2,3,4,5,6,7,8,9), nrow=3, byrow=FALSE)
     [,1] [,2] [,3]
[1,]    1    4    7
[2,]    2    5    8
[3,]    3    6    9
```

注意參數的擺放位置和用法 {nrow} {byrow}

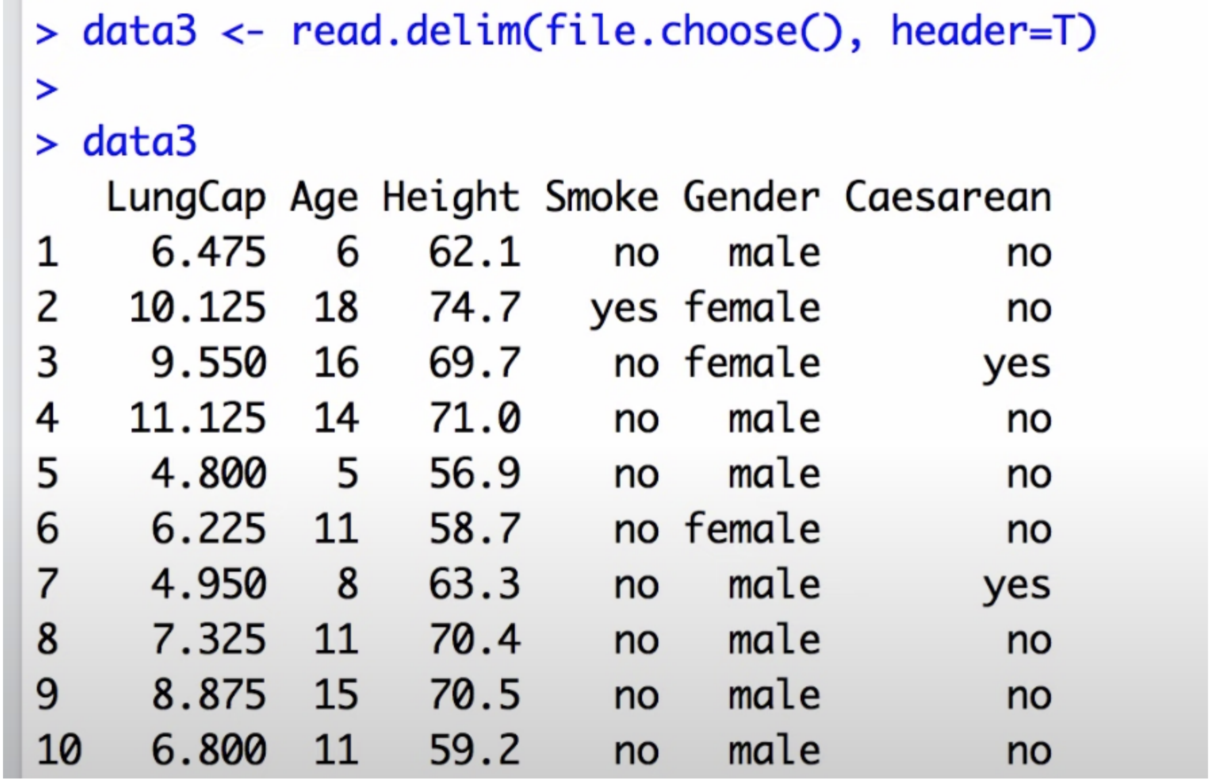
Comma separated value: .csv
V.s
Tab delimited text file; .txt



```
> help(read.csv)
> ?read.csv
>
> data1 <- read.csv(file.choose(), header=T)
>
> data1
   LungCap Age Height Smoke Gender Caesarean
1    6.475   6   62.1    no   male        no
2   10.125  18   74.7   yes female        no
3    9.550  16   69.7    no female       yes
4   11.125  14   71.0    no   male        no
5    4.800   5   56.9    no   male        no
6    6.225  11   58.7    no female        no
7    4.950   8   63.3    no   male       yes
8    7.325  11   70.4    no   male        no
9    8.875  15   70.5    no   male        no
10   6.800  11   59.2    no   male        no
```

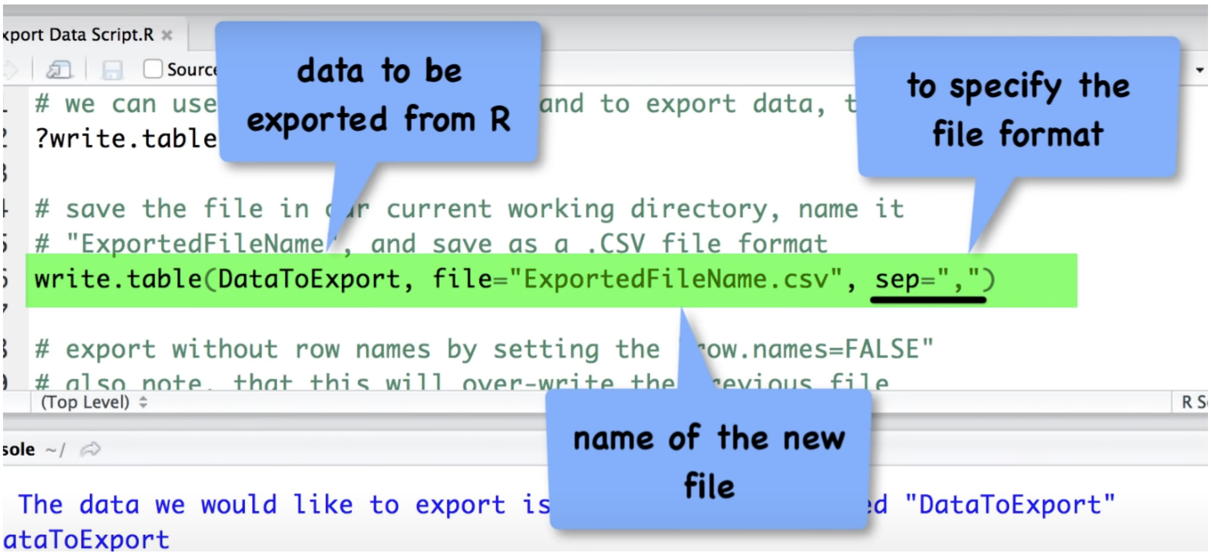


```
> data2 <- read.table(file.choose(), header=T, sep=",")
>
> data2
   LungCap Age Height Smoke Gender Caesarean
1    6.475   6   62.1    no   male        no
2   10.125  18   74.7   yes female        no
3    9.550  16   69.7    no female       yes
4   11.125  14   71.0    no   male        no
5    4.800   5   56.9    no   male        no
6    6.225  11   58.7    no female        no
7    4.950   8   63.3    no   male       yes
8    7.325  11   70.4    no   male        no
9    8.875  15   70.5    no   male        no
10   6.800  11   59.2    no   male        no
```



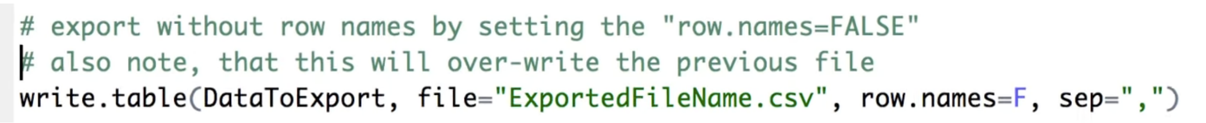
```
> data3 <- read.delim(file.choose(), header=T)
>
> data3
   LungCap Age Height Smoke Gender Caesarean
1    6.475   6   62.1    no   male        no
2   10.125  18   74.7   yes female        no
3    9.550  16   69.7    no female       yes
4   11.125  14   71.0    no   male        no
5    4.800   5   56.9    no   male        no
6    6.225  11   58.7    no female        no
7    4.950   8   63.3    no   male       yes
8    7.325  11   70.4    no   male        no
9    8.875  15   70.5    no   male        no
10   6.800  11   59.2    no   male        no
```

1.6 Export Data from R (csv , txt and other formats)

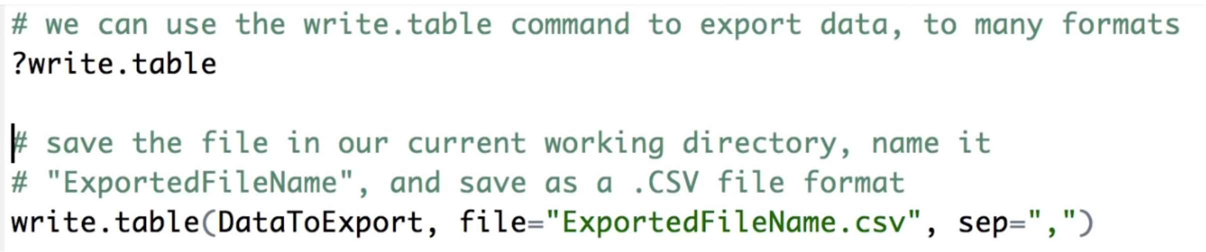


```
# we can use                  and to export data, t
?write.table

# save the file in our current working directory, name it
# "ExportedFileName", and save as a .CSV file format
write.table(DataToExport, file="ExportedFileName.csv", sep=",")

# export without row names by setting the  row.names=FALSE"
# also note, that this will over-write the  revious file
```

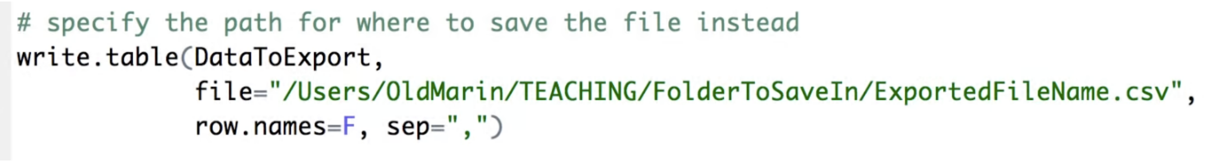
```
 The data we would like to export is                   d "DataToExport"
ataToExport
```



```
# export without row names by setting the "row.names=FALSE"
# also note, that this will over-write the previous file
write.table(DataToExport, file="ExportedFileName.csv", row.names=F, sep=",")
```

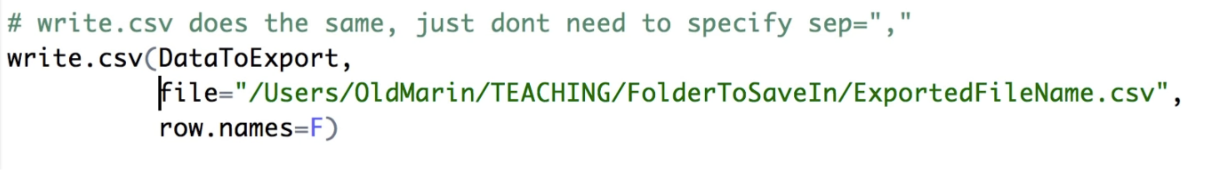


```
# we can use the write.table command to export data, to many formats
?write.table

# save the file in our current working directory, name it
# "ExportedFileName", and save as a .CSV file format
write.table(DataToExport, file="ExportedFileName.csv", sep=",")
```



```
# specify the path for where to save the file instead
write.table(DataToExport,
            file="/Users/OldMarin/TEACHING/FolderToSaveIn/ExportedFileName.csv",
            row.names=F, sep=",")
```

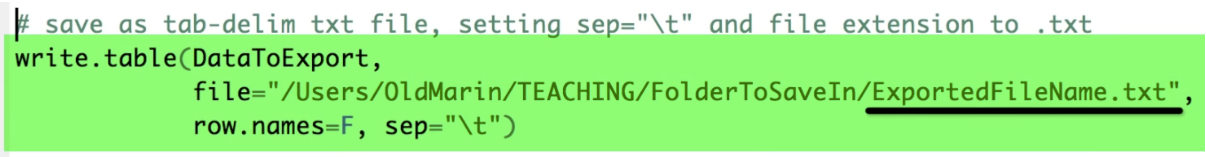
Save to new directory



```
# write.csv does the same, just dont need to specify sep=","
write.csv(DataToExport,
          file="/Users/OldMarin/TEACHING/FolderToSaveIn/ExportedFileName.csv",
          row.names=F)
```

To export data from R into comma separated value file European style use: **write.csv2**
This will use comma for decimal points and semi-colon for separator

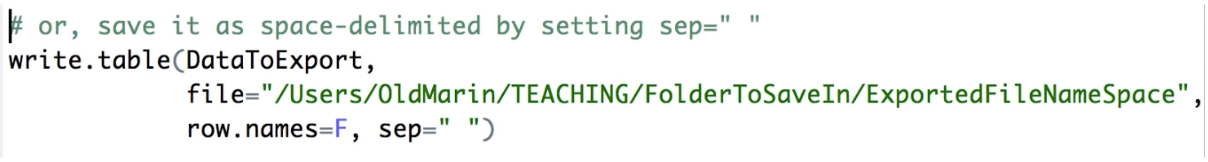
Write.table command allows us to save the file in other formats, (ex : tab_delimited text file)



```
# save as tab-delim txt file, setting sep="\t" and file extension to .txt
write.table(DataToExport,
            file="/Users/OldMarin/TEACHING/FolderToSaveIn/ExportedFileName.txt",
            row.names=F, sep="\t")
```

sep= “\t” -> for tab-delimited file

The file can also be saved in “ “



```
# or, save it as space-delimited by setting sep=" "
write.table(DataToExport,
            file="/Users/OldMarin/TEACHING/FolderToSaveIn/ExportedFileNameSpace",
            row.names=F, sep=" ")
```

```
LungCapdata <- read.csv(file.choose(), header=T, sep=",")
# Read in Data
# attach Data
attach(LungCapdata)


#Check the names
names(LungCapdata)
# Check the first 6 rows
head(LungCapdata)

Age[1:5] # find the age for fisrt 5 individuals
# Check whether the age is greater than 15
temp <- Age>15
temp[1:5]
# add numeric variable
temp2 <- as.numeric(Age>15)
# Look first 5 observation for all of our data
LungCapdata[1:5, ]
# Try to use multiple logical statements within an R command to have a logical vector
# answering multiple questions

#that's try to find female who smoke
Femsmoke <- Gender=="female" & Smoke=="yes"
Femsmoke[1:5]

# We can attach data in a column-wise fashion using "cbind"
# Can also attach them in a row-wise fashion using "rbind"
MoreData <- cbind(LungCapdata, Femsmoke)
MoreData[1:5, ]



# Video 11: Setting up Working Directory in R
# A working directory is one spot that you have created for saving all of your works

# Find what the current working directory
getwd()

# Can set working directory in few different ways
#1. using"setwd()"
```

```
setwd("/Users/oliviawang/Desktop/R language") # specify the path to the folder
# Don't need to use full path, can use "~"
setwd("~/Desktop/R language")

# 2.save.image() save the current image
# first create some itmes
MeanAge = mean(Age)
x = c(1,2,3,4,5)
y = 14
z = summary(LungCapdata)
# then save it
save.image("projectone.Rdata")
# without specifying a path, it will be saved in the current working directory

#now, remove all items from the current workspace
rm(list=ls())

# tutorial 12: Writing scripts of code in R

# A script is a set of commands that usually includes commenting
# on what each piece of code is intended to do
class(Gender)

#use tab to find suggestion

# Video 13: installing package in R
help(install.package)
install.packages("epiR")

# check all available packages
install.packages()

# get help
help(package=epiR)
```